

Generation of Rules from Incomplete Information Systems

Marzena Kryszkiewicz
Institute of Computer Science
Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: mkr@ii.pw.edu.pl

Abstract. In the paper we define certain and possible rules in an incomplete information system as certain/possible ones in every completion of the initial system. The careful examination of the dependencies between an incomplete system and its completions allow us to state that it is feasible to generate all certain rules and some important class of possible rules directly from the incomplete information system. Space complexity of the proposed method of rules' generation is linear with regard to the number of objects in the initial system.

1 Introduction

The problem of knowledge discovering in the form of rules from incomplete information systems (*IS*) is considered. By an incomplete system we mean a system with missing data (null values). We do not consider the case of null value meaning *inapplicable* value. This problem may be solved by adding a special symbol denoting inapplicable value to the attribute domains. In the paper we deal with the problem of *unknown* values.

Several solutions to the problem of generating decision tree from the training set of examples with unknown values have been proposed in the area of Artificial Intelligence. The simplest ones consist in removing examples with unknown values or replacing unknown values with the most common values. More complex approaches were presented in [1-2]. A Bayesian formalism is used in [1] to determine the probability distribution of the unknown value over the possible values from the domain. This method could either choose the most likely value or divide the object into fractional objects, each with one possible value weighted according to the probabilities determined. It is suggested in [2] to predict the value of an attribute based on the value of other attributes of the object, and the class information.

The problem of rules' generation from incomplete systems was investigated also in the context of Rough Sets [3-6]. The methodology from [3] consists in transforming an incomplete system to a complete system, where each object with incomplete descriptor in the source system is represented by a set of possible subobjects in the target system. As we prove in [4], this method allows to generate the set of all certain rules. Different methodology was presented in [5]. The method considered in [5] allows to generate a subset of certain rules directly from the original incomplete decision table. Modelling uncertainty caused by appearance of unknown values by means of fuzzy sets was discussed in [6].

In this paper we refer to the results we obtained in [4-5] and extend them so that the generation of all rules and some important class of possible rules is feasible. The unique feature of certain/possible rules we consider is that they are certain/possible in every completion of the initial *IS*. The careful examination of the dependencies

between an incomplete IS and its completions allow us to state that such definitions of decision rules do not require performing the calculations in all the completions of the incomplete IS . We propose a method of generation of all rules directly from the original incomplete IS . Space complexity of the method is $O(n)$, where n is the number of objects in the initial system.

2 Information Systems

Information system (IS) is a triplet $\mathcal{S} = (\mathcal{O}, AT, f)$, where \mathcal{O} - is a non-empty finite set of objects and AT is a non-empty finite set of attributes, such that $f_a: \mathcal{O} \rightarrow V_a$ for any $a \in AT$, where V_a is called *domain* of an attribute a . Any attribute domain V_a may contain special symbol "*" to indicate that the value of an attribute is *unknown (null)*. Here, we assume that an object $x \in \mathcal{O}$ possesses only one value for an attribute $a, a \in AT$, in reality. Thus, if the value of an attribute a is missing then we may conclude that the real value must be one from the set $V_a \setminus \{*\}$. System in which values of all attributes for all objects from \mathcal{O} are known is called *complete*, otherwise it is called *incomplete*.

Let $\mathcal{S}' = (\mathcal{O}, AT, f')$. We say that \mathcal{S}' is an *extension* of \mathcal{S} if \mathcal{S}' is an IS such that if $f_a(x) \neq *$ then $f_a'(x) = f_a(x)$ for all $a \in AT$ and $x \in X$. We say that \mathcal{S}' is a *completion* of \mathcal{S} if \mathcal{S}' is a complete IS , which is an extension of \mathcal{S} .

Let us note that if $card(V_a) = 1$ then null value occurring for attribute a may be substituted by the unique domain value $v_a \in V_a$ without changing information capabilities of the system. Hence, any system \mathcal{S} is equivalent to its extension \mathcal{S}' such that if $f_a(x) = *$ then $f_a'(x) = v_a, v_a \in V_a$, for each attribute $a \in AT$ such that $card(V_a) = 1$ and for each $y \in \mathcal{O}$. From now on, without loss of generality, we will consider only systems that do not have null values for attributes a such that $card(V_a) = 1$.

In the sequel, any attribute-value pair $(a, v), a \in AT, v \in V_a$, will be called an *atomic property*. Any *atomic property* or its conjunction will be called *descriptor*. Conjunction of atomic properties for all attributes AT will be called *full descriptor*. Descriptor that does not possess null values for attributes $A \subseteq AT$ will be called *A-complete*. The set of objects having the atomic property (a, v) , i.e. $\{x \in \mathcal{O} \mid f_a(x) = v\}$, will be denoted by $\|(a, v)\|$. Let us note that $\|(a, *)\| \cap \|(a, v)\| = \emptyset$, if $v \neq *$. The set of objects satisfying any descriptor t will be denoted by $\|t\|$ and will be computed in the usual way, e.g. $\|t \wedge s\| = \|t\| \cap \|s\|$.

3 Indiscernibility of Objects and Set Approximations

3.1 Indiscernibility of Objects

Let $\mathcal{S} = (\mathcal{O}, AT, f)$. Each subset of attributes $A \subseteq AT$ determines a binary *indiscernibility relation* $IND(A)$, $IND(A) = \{(x, y) \in \mathcal{O} \times \mathcal{O} \mid \forall a \in A, f_a(x) = f_a(y)\}$. The relation $IND(A)$, $A \subseteq AT$, is an equivalence relation and constitutes a partition of \mathcal{O} , which we will denote by $\mathcal{O} / IND(A)$. Let $I_A(x)$ denote the object set $\{y \in \mathcal{O} \mid (x, y) \in IND(A)\}$. Objects from $I_A(x)$ are indiscernible with regard to their description in the system, but they may have different properties in reality, unless the system is complete. Objects perceived as indiscernible in the complete system are indiscernible also in reality.

Another *similarity relation* $SIM(A)$, $SIM(A)=\{(x,y)\in O\times O \mid \forall a\in A, f_a(x)=f_a(y) \text{ or } f_a(x)=* \text{ or } f_a(y)=*\}$, treats two objects as similar if they may have the same properties in reality. Similarity relation is reflexive and symmetric, but may not be transitive. By $S_A(x)$ we will denote the set of objects $\{y\in O \mid (x,y)\in SIM(A)\}$.

Property 3.1.1

- Let $B \subseteq A \subseteq AT$. Then: $I_A(x) \subseteq I_B(x)$; $S_A(x) \subseteq S_B(x)$; $I_A(x) \subseteq S_A(x)$.
- Let \mathcal{S}' be a completion of \mathcal{S} . Then: $I_A'(x) = S_A'(x)$; $S_A'(x) \subseteq S_A(x)$.
- $S_A(x) = \bigcup_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} S_A'(x)$.

3.2 Set Approximations

Let $X \subseteq O$ and $A \subseteq AT$. First we recall standard rough set definitions of *lower approximation* $\underline{A}_{IND}X$ and *upper approximation* $\overline{A}_{IND}X$ of X in a complete IS :

$$\underline{A}_{IND}X = \{x \in O \mid I_A(x) \subseteq X\} = \{x \in X \mid I_A(x) \subseteq X\},$$

$$\overline{A}_{IND}X = \{x \in O \mid I_A(x) \cap X \neq \emptyset\} = \bigcup \{I_A(x) \mid x \in X\}.$$

Beneath we generalize these definitions for the case of incomplete IS :

$$\underline{A}_{SIM}X = \{x \in O \mid S_A(x) \subseteq X\} = \{x \in X \mid S_A(x) \subseteq X\},$$

$$\overline{A}_{SIM}X = \{x \in O \mid S_A(x) \cap X \neq \emptyset\} = \bigcup \{S_A(x) \mid x \in X\}.$$

$\underline{A}_{SIM}X$ is a set of objects that belong to X with certainty, while $\overline{A}_{SIM}X$ is a set of objects that possibly belong to X . Obviously, $\underline{A}_{SIM}X = \underline{A}_{IND}X$ and $\overline{A}_{SIM}X = \overline{A}_{IND}X$ in a complete IS . $\underline{A}_{SIM}X$ and $\overline{A}_{SIM}X$ will be denoted also by $\underline{A}X$ and $\overline{A}X$, respectively.

Property 3.2.1

- Let \mathcal{S}' be an extension of \mathcal{S} and $X \subseteq O$ and $A \subseteq AT$. Then: $X \subseteq \overline{A}X' \subseteq \overline{A}X$.
- $\bigcup_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{A}X)' = \overline{A}X$.

Property 3.2.2

Let $A^*(y)$ denote $\{a \in A \mid f_a(y) = *\}$ and $L_{\max}(y)$ denote $\prod_{a \in A^*(y)} \text{card}(V_a)$. Let $Full(Y)$ denote

the set of all objects from Y which have full complete descriptors. Let $Desc(Y)$ denote the set of all different descriptors of objects Y .

$$\bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{A}X)' = (\overline{A}X) \setminus H, \text{ where } X \subseteq O \text{ and}$$

$$H = \{y \in (\overline{A}X \setminus X) \mid \text{card}(Desc(Full(S_A(y) \cap X))) < L_{\max}(y)\}.$$

Proof:

By Property 3.2.1 $\overline{A}X$ is the set of all objects $y \in O$ such that $y \in (\overline{A}X)'$ in some completion \mathcal{S}' of \mathcal{S} . On the other hand, H is a set of objects $y \in (\overline{A}X) \setminus X$ satisfying the condition $\text{card}(Desc(Full(S_A(y) \cap X))) < L_{\max}(y)$, which implies that $y \notin (\overline{A}X)'$ in some completion \mathcal{S}'' of \mathcal{S} . Hence, H is the set of all objects from $\overline{A}X$ that do not belong to

$$\bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{A}X)'. \text{ Thus, the difference between } \overline{A}X \text{ and } H \text{ is equal to } \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{A}X)'.$$

□

Property 3.2.3

Let $X \subseteq \mathcal{O}$ and $A \subseteq AT$. Let \mathcal{S}' and \mathcal{S}'' be completions of \mathcal{S} which differ only for an object $x \in \mathcal{O}$. If $S_A(x) \subseteq \bigcap_{\mathcal{S}''' \text{ is a completion of } \mathcal{S}} (\overline{AX})'''$ then $(\overline{AX})' = (\overline{AX})''$.

Proof:

Since $S_A(x) \subseteq \bigcap_{\mathcal{S}''' \text{ is a completion of } \mathcal{S}} (\overline{AX})'''$ then $S_A'(x) \subseteq (\overline{AX})''$. Additionally, $S_A'(x) \cap S_A''(x) = \{x\}$

because completions \mathcal{S}' and \mathcal{S}'' differ only for x . Hence, $S_A'(x) \setminus \{x\} \subseteq (\overline{AX})'' \setminus S_A''(x) \subseteq (\overline{A(X \setminus \{x\})})''$. Similarly, we may derive: $S_A''(x) \setminus \{x\} \subseteq (\overline{A(X \setminus \{x\})})'$.

We may also notice that $\overline{S_A'(y)} = S_A''(y)$ or $\overline{S_A'(y)} = S_A''(y) \setminus \{x\}$ or $\overline{S_A'(y)} = S_A''(y) \cup \{x\}$ for any $y \in X \setminus \{x\}$. Hence: $(\overline{A(X \setminus \{x\})})' \cup \{x\} = (\overline{A(X \setminus \{x\})})'' \cup \{x\}$.

Applying the results obtained above, we may write: $(\overline{AX})'' = S_A''(x) \cup (\overline{A(X \setminus \{x\})})''$
 $= (S_A''(x) \setminus \{x\}) \cup \{x\} \cup (\overline{A(X \setminus \{x\})})'' \cup (S_A'(x) \setminus \{x\}) = (S_A''(x) \setminus \{x\}) \cup \{x\} \cup (\overline{A(X \setminus \{x\})})' \cup (S_A'(x) \setminus \{x\}) = \{x\} \cup (\overline{A(X \setminus \{x\})})' \cup (S_A'(x) \setminus \{x\}) = (\overline{A(X \setminus \{x\})})' \cup S_A'(x) = (\overline{AX})'$. \square

4 Complete Decision Tables

Decision table (DT) is an information system $\mathcal{S} = (\mathcal{O}, AT \cup \{d\}, f)$, where $d, d \notin AT$ and $* \notin V_d$ is a distinguished attribute called *decision*, and the elements of AT are called *conditions*. If DT is a complete IS then it is called *complete decision table*, otherwise it is called *incomplete decision table*.

In Section 4 we will restrict our considerations only to complete decision tables.

4.1 Decision rules

Decision rules we will consider will have the form: $t \rightarrow s$, where $t = \bigwedge (c, v)$, $c \in A \subseteq AT$, $v \in V_c$, and $s = (d, w)$, $w \in V_d$. In the sequel, we will call t and s *condition* and *decision part* of a rule, respectively. We will say that object x , $x \in \mathcal{O}$, *supports* a rule $t \rightarrow s$ in \mathcal{S} (or a rule $t \rightarrow s$ *covers* an object x in \mathcal{S}) if x has both property t and s in \mathcal{S} . A decision rule $t \rightarrow s$ is *certain* in \mathcal{S} if $\|t\| \subseteq \|s\|$ in \mathcal{S} . A decision rule $t \rightarrow s$ is *possible* in \mathcal{S} if $\|t\| \subseteq \overline{AT} \|s\|$ in \mathcal{S} . A certain (possible) decision rule $t \rightarrow s$ is *optimal* in \mathcal{S} iff it is certain (possible) in \mathcal{S} and no other rule constructed from a proper subset of atomic properties occurring in t is certain (possible) in \mathcal{S} .

Property 4.1.1

If object x , $x \in \mathcal{O}$, supports a certain rule in \mathcal{S} then $I_{AT}(x) \subseteq I_{\{d\}}(x)$.

4.2 Reducts

Here we will present a method of computing all optimal rules supported by an arbitrary object $x \in \mathcal{O}$. Let $t \rightarrow s$ be an optimal rule supported by object $x \in \mathcal{O}$. The decision part s is determined by the decision $d(x)$. The condition part t is a conjunction of atomic properties of x for attributes A , where A is the set of all attributes occurring in t . Since $t \rightarrow s$ is optimal A is a minimal attribute set for which $t \rightarrow s$ is certain or possible, respectively. Any set A of this property will be called a *reduct* for x . Below we provide formal definitions of a certain and possible reduct:

Set $A, A \subseteq AT$ is a *certain reduct* for $x, I_{AT}(x) \subseteq I_{\{a\}}(x)$, in \mathcal{S} , iff A is a minimal set such that $I_A(x) \subseteq I_{\{a\}}(x)$. Set $A, A \subseteq AT$ is a *possible reduct* for $x, x \in \mathcal{O}$, in \mathcal{S} , iff A is a minimal set such that $I_A(x) \subseteq \overline{AT}(I_{\{a\}}(x))$.

In general, object x may serve as a generator of all optimal rules such that their conditional parts are determined by reducts for x and their decision part is equal to $(d, d(x))$.

In order to compute reducts of DT we will exploit the idea of so called *discernibility functions* [7]. Their main properties are that they are monotonic Boolean functions and their prime implicants determine reducts uniquely.

Let $\alpha_{A}(x, y)$ be a set of attributes $a \in A$ such that $(x, y) \notin SIM(\{a\})$. Let $\sum \alpha_{A}(x, y)$ be equal to 1, if $\alpha_{A}(x, y) = \emptyset$. Otherwise, let $\sum \alpha_{A}(x, y)$ be a disjunction of Boolean variables corresponding to attributes contained in $\alpha_{A}(x, y)$.

$\Delta_c(x)$ is a *certain discernibility function* for object $x, I_{AT}(x) \subseteq I_{\{a\}}(x)$, iff

$$\Delta_c(x) = \prod_{y \in Y_c} \sum \alpha_{AT}(x, y), \text{ where } Y_c = \mathcal{O} \setminus I_{\{a\}}(x).$$

$\Delta_p(x)$ is a *possible discernibility function* for object $x, x \in \mathcal{O}$, iff

$$\Delta_p(x) = \prod_{y \in Y_p} \sum \alpha_{AT}(x, y), \text{ where } Y_p = \mathcal{O} \setminus \overline{AT}(I_{\{a\}}(x)).$$

5 Incomplete Decision Table

5.1 Certain Rules

Following the approach to incomplete information systems presented by Lipski in [8], we propose the following definition of a certain rule: Rule $t \rightarrow s$ is *certain* in \mathcal{S} if it is certain in every completion of \mathcal{S} . A certain decision rule $t \rightarrow s$ is *optimal* iff it is certain in \mathcal{S} and no other rule constructed from a proper subset of atomic properties occurring in t is certain. Object $x \in \mathcal{O}$ *supports* a certain rule $t \rightarrow s$ in \mathcal{S} iff x supports $t \rightarrow s$ in each completion of \mathcal{S} in which $\|t\| \neq \emptyset$.

Property 5.1.1

Rule $t \rightarrow s$ is certain in \mathcal{S} if it is certain in every completion of \mathcal{S} in which $\|t\| \neq \emptyset$.

Proposition 5.1.1

Rule $t \rightarrow s$ is certain in \mathcal{S} if there exists an object $x \in \|t\|$ in some completion of \mathcal{S} and $t \rightarrow s$ is certain in all completions of \mathcal{S} in which $x \in \|t\|$.

Proposition 5.1.2

Let \mathcal{S}^e be an extension of \mathcal{S} such that there is an object $x \in \|t\|$ having full complete descriptor in \mathcal{S}^e and \mathcal{S}^e may differ from \mathcal{S} only for object x .

Rule $t \rightarrow s$ is certain in \mathcal{S} if it is certain in \mathcal{S}^e .

Property 5.1.2

Let \mathcal{S}^e be an extension of \mathcal{S} such that there is an object $x \in \|t\|$ having full complete descriptor in \mathcal{S}^e and \mathcal{S}^e may differ from \mathcal{S} only for object x .

If object $x, x \in \mathcal{O}$, supports a certain rule in \mathcal{S}^e then $S_{AT}^e(x) \subseteq I_{\{a\}}(x)$.

Proposition 5.1.3

Let \mathcal{S}^e be an extension of \mathcal{S} such that there is an object $x \in \mathcal{O}$ having full complete descriptor in \mathcal{S}^e and $S_{AT}^e(x) \subseteq I_{\{a\}}(x)$ and \mathcal{S}^e may differ from \mathcal{S} only for object x .

Any rule $t \rightarrow s$ is certain in \mathcal{S} if $t \rightarrow s$ is certain and supported by x in \mathcal{S}^e .

Proof: Immediate from Proposition 5.1.2 and Property 5.1.2. □

5.2 Computation of Certain Reducts

It is stated in Proposition 5.1.3 that certain rules may be computed in any extension \mathcal{S}^e of \mathcal{S} such that the values of all attributes are the same in \mathcal{S} and \mathcal{S}^e for all objects from $\mathcal{O} \setminus \{x\}$ and x has full complete descriptor in \mathcal{S}^e and $S_{AT}^e(x) \subseteq I_{\{d\}}(x)$. The number of different extensions \mathcal{S}^e is equal to the number of different substitutions for null values in x .

In order to compute the condition attributes of an optimal certain rule the notion of a *certain reduct* in \mathcal{S}^e may be exploited: Set A , $A \subseteq AT$ is a *certain reduct* for x , $S_{AT}^e(x) \subseteq I_{\{d\}}(x)$, in system \mathcal{S}^e iff A is a minimal set such that $I_{AT'}(x) \subseteq I_{\{d\}}(x)$ in every completion \mathcal{S}' of \mathcal{S}^e . The set of all certain reducts of x in \mathcal{S} is equal to the set-theoretical sum of all certain reducts for x computed in all \mathcal{S}^e .

$\Delta_c^e(x)$ is a *certain discernibility function* for object x in \mathcal{S}^e , $S_{AT}^e(x) \subseteq I_{\{d\}}(x)$, iff

$$\Delta_c^e(x) = \prod_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} \prod_{y \in Y_c} \sum \alpha_{AT'}(x, y), \text{ where } Y_c = \mathcal{O} \setminus I_{\{d\}}(x).$$

All certain reducts of x in a \mathcal{S}^e may be computed as all prime implicants of $\Delta_c^e(x)$.

Proposition 5.2.1

$\Delta_c^e(x)$ is a certain discernibility function for object x in \mathcal{S}^e , $S_{AT}^e(x) \subseteq I_{\{d\}}(x)$, if

$$\Delta_c^e(x) = \prod_{y \in Y_c} \sum \alpha_{AT}^e(x, y), \text{ where } Y_c = \mathcal{O} \setminus I_{\{d\}}(x).$$

Proof:

Let \mathcal{S}' be a completion of \mathcal{S}^e such that all null values occurring in \mathcal{S}^e are replaced in \mathcal{S}' with the respective attribute values occurring in the descriptor of x . Let $\mathcal{S}'' \neq \mathcal{S}'$ be an arbitrary completion of \mathcal{S}^e . One may easily notice that for each $y \in \mathcal{O} \setminus I_{\{d\}}(x)$, $\sum \alpha_{A'}(x, y) \wedge \sum \alpha_{A''}(x, y) = \sum \alpha_{A'}(x, y) = \sum \alpha_{A}^e(x, y)$. Since \mathcal{S}'' was chosen arbitrarily then we may generalize the above observation:

$$\Delta_c^e(x) = \prod_{\mathcal{S}'' \text{ is a completion of } \mathcal{S}^e} \prod_{y \in \mathcal{O} \setminus I_{\{d\}}(x)} \sum \alpha_{AT''}(x, y) = \prod_{y \in \mathcal{O} \setminus I_{\{d\}}(x)} \sum \alpha_{AT'}(x, y) = \prod_{y \in \mathcal{O} \setminus I_{\{d\}}(x)} \sum \alpha_{AT}^e(x, y). \quad \square$$

5.3 Possible Rules

Rule $t \rightarrow s$ is *possible* in \mathcal{S} if it is possible in every completion of \mathcal{S} . A possible decision rule $t \rightarrow s$ is *optimal* iff it is possible in \mathcal{S} and no other rule constructed from a proper subset of atomic properties occurring in t is possible. Object $x \in \mathcal{O}$ *supports* a possible rule $t \rightarrow s$ in \mathcal{S} iff x supports $t \rightarrow s$ in each completion of \mathcal{S} in which $\|t\| \neq \emptyset$.

Property 5.3.1

Rule $t \rightarrow s$ is possible in \mathcal{S} if it is possible in every completion of \mathcal{S} in which $\|t\| \neq \emptyset$.

Proposition 5.3.1

Let \mathcal{S}^e be an extension of \mathcal{S} such that there is an object $x \in \|t\|$ in \mathcal{S}^e and $S_{AT}(x) \subseteq X$, where $X = \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{AT \|s\|})'$, and \mathcal{S}^e may differ from \mathcal{S} only for object x .

The rule $t \rightarrow s$ is possible in \mathcal{S} if it is possible in \mathcal{S}^e .

Proof:

Let $F(z,t)$ denote a set of all completions of \mathcal{S} in which $z \in \|t\|$. Let x and y be arbitrary objects such that $x \neq y$ and $F(x,t) \neq \emptyset$ and $F(y,t) \neq \emptyset$. Additionally, we assume that rule $t \rightarrow s$ is possible in all completions from $F(x,t)$. Let \mathcal{S}' be an arbitrary completion of \mathcal{S} such that $\mathcal{S}' \in F(y,t) \setminus F(x,t)$. Obviously, $x \notin \|t\|$. On the other hand there is a completion \mathcal{S}'' in $F(x,t)$ such that \mathcal{S}'' differs from \mathcal{S}' only for object x . This implies that $x \in \|t\|$. Since $\|t\| \subseteq (\overline{AT\|s\|})$ and $\|t\| \cup \{x\} = \|t\|$ and $(\overline{AT\|s\|})' = (\overline{AT\|s\|})$ (by Property 3.2.3), then $\|t\| \subseteq (\overline{AT\|s\|})' = (\overline{AT\|s\|})$. Hence, rule $t \rightarrow s$ which is possible in \mathcal{S}'' is also possible in \mathcal{S}' .

Since our choice of \mathcal{S}' was arbitrary we may conclude that if rule $t \rightarrow s$ is possible in all completions $F(x,t)$, then it is also possible in all completions $F(y,t)$. Additionally, since we chose x and y arbitrarily, we can infer that if rule $t \rightarrow s$ is possible in all systems $F(x,t) \neq \emptyset$ generated by any object x supporting rule $t \rightarrow s$, then it is possible in all completions $F(y,t)$ generated by all objects y such that $F(y,t) \neq \emptyset$. Proposition 5.3.1 is an immediate consequence of this conclusion and Property 5.3.2. \square

Proposition 5.3.2

Let \mathcal{S}^e be an extension of \mathcal{S} such that there is an object $x \in \|t\|$ having full complete descriptor in \mathcal{S}^e and $S_{AT}(x) \subseteq X$, where $X = \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{AT\|s\|})'$, and \mathcal{S}^e may differ from \mathcal{S}

only for object x .

The rule $t \rightarrow s$ is possible in \mathcal{S} if it is possible in \mathcal{S}^e .

Proof:

Let x be an object having full complete descriptor and $x \in \|t\|$ in \mathcal{S}^e . Let $F(x,t)$ be a set of all completions of \mathcal{S} in which $x \in \|t\|$. Let $G(x,t)$ be a set of all completions of \mathcal{S}^e having the same descriptor of object x . Since $x \in \|t\|$ in \mathcal{S}^e then $G(x,t) \neq \emptyset$ and $G(x,t) \subseteq F(x,t) \neq \emptyset$. Additionally, we assume that rule $t \rightarrow s$ is possible in \mathcal{S}^e , i.e. $t \rightarrow s$ is possible in all completions from $G(x,t)$. Let \mathcal{S}' be an arbitrary completion of \mathcal{S} such that $\mathcal{S}' \in F(x,t) \setminus G(x,t)$. There is a completion \mathcal{S}'' in $G(x,t)$ such that \mathcal{S}'' differs from \mathcal{S}' only for object x . However, $\|t\|' = \|t\|$ and $(\overline{AT\|s\|})' = (\overline{AT\|s\|})$ (by Property 3.2.3), so $\|t\| \subseteq (\overline{AT\|s\|})$ implies $\|t\| \subseteq (\overline{AT\|s\|})'$. Hence, rule $t \rightarrow s$ which is possible in \mathcal{S}'' is also possible in \mathcal{S}' . Since our choice of \mathcal{S}' was arbitrary we may conclude that if rule $t \rightarrow s$ is possible in all completions $G(x,t)$ (i.e. if rule $t \rightarrow s$ is possible in \mathcal{S}^e), then it is also possible in all systems $F(x,t)$. Additionally, since we chose \mathcal{S}^e and x arbitrarily, we can infer that if rule $t \rightarrow s$ is possible in \mathcal{S}^e , then it is possible in all completions $F(x,t)$ and $F(x,t) \neq \emptyset$. Proposition 5.3.3 is an immediate consequence of this conclusion and Proposition 5.3.2. \square

Proposition 5.3.3

Let \mathcal{S}^e be an extension of \mathcal{S} such that there is an object $x \in \emptyset$ having full complete descriptor in \mathcal{S}^e and $S_{AT}(x) \subseteq \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI_d}(x))'$ and \mathcal{S}^e may differ from \mathcal{S} only for x .

Any rule $t \rightarrow s$ is possible in \mathcal{S} if $t \rightarrow s$ is possible and supported by x in \mathcal{S}^e .

Proof: Immediate from Proposition 5.3.2. □

5.4 Computation of Possible Reducts

Let \mathcal{S}^e be an extension of \mathcal{S} such that \mathcal{S}^e may differ from \mathcal{S} only for some $x \in \mathcal{O}$. In this subsection we will show how to compute possible rules supported by object x having full descriptor in \mathcal{S}^e and such that $S_{AT}(x) \subseteq \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI_d(x)})'$. There may be several

extensions \mathcal{S}^e of \mathcal{S} for a fixed x . Our approach is justified by Proposition 5.3.3. In the remainder of this subsection, Y_p (perhaps accompanied by an index referring to the respective system) will denote $\mathcal{O} \setminus \overline{ATI_{\{a\}}}(x)$.

In order to compute condition part of an optimal possible rule the notion of a *possible reduct* in \mathcal{S} may be exploited. Set A , $A \subseteq AT$ is a *possible reduct* for x , where $S_{AT}(x) \subseteq \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI_d(x)})'$, in system \mathcal{S}^e iff A is a minimal set such that

$I_A(x) \subseteq \overline{ATI_{\{a\}}}(x)$ in every completion \mathcal{S}' of \mathcal{S}^e . The set of all possible reducts of x in \mathcal{S} is equal to the set-theoretical sum of all possible reducts for x computed for all \mathcal{S}^e .

$\Delta_p^e(x)$ is a *possible discernibility function* for object x , $S_{AT}(x) \subseteq \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI_d(x)})'$, in \mathcal{S}^e , iff

$$\Delta_p^e(x) = \prod_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} \prod_{y \in Y_p} \alpha_{AT}'(x, y).$$

All possible reducts of x in \mathcal{S}^e may be computed as all prime implicants of $\Delta_p^e(x)$.

Proposition 5.4.1

Let \mathcal{S}^x be a completion of \mathcal{S}^e such that all null values occurring in \mathcal{S}^e are replaced in \mathcal{S}^x with respective attribute values characterizing object x . Let $\beta_A(x, y)$ be a set of attributes $a \in A$ such that $a(x) \neq a(y)$ or $a(x) = *$ or $a(y) = *$ in \mathcal{S}^e .

$\Delta_p^e(x)$ is a *possible discernibility function* for object x , $S_{AT}(x) \subseteq \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI_d(x)})'$, in \mathcal{S}^e , iff

$$\Delta_p^e(x) = \prod_{y \in \overline{V}_p} \beta_{AT}^e(x, y) \wedge \prod_{y \in \overline{V}_p^x} \alpha_{AT}^e(x, y), \text{ where } Y = \mathcal{O} \setminus \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} (\overline{ATI_d(x)})'.$$

Proof:

Let $* \neq$ denote a special value which is different from any domain value. Let \mathcal{S}^s be an IS such that all null values occurring in \mathcal{S}^e for all objects from X are replaced in \mathcal{S}^s with respective attribute values characterizing x and all the other missing values in \mathcal{S}^s are replaced by special value $* \neq$. Let us consider five types of completions of \mathcal{S}^e :

- 1) \mathcal{S}' is a completion of \mathcal{S}^e such that all null values occurring in \mathcal{S}^e for all objects from $\mathcal{O} \setminus V_{\{a\}}(x)$ are replaced in \mathcal{S}' with arbitrary attribute values different from those characterizing x and the other null values are replaced by respective attribute values characterizing object x .
- 2) \mathcal{S}' is a completion of \mathcal{S}^e such that all null values occurring in \mathcal{S}^e for all objects from $\mathcal{O} \setminus V_{\{a\}}(x)$ are replaced in \mathcal{S}' with arbitrary attribute values different from those characterizing x .

- 3) \mathcal{S}' is a completion of \mathcal{S}^e such that all null values occurring in \mathcal{S}^e are replaced in \mathcal{S}' with respective attribute values characterizing object x .
- 4) \mathcal{S}' is a completion of \mathcal{S}^e such that null values occurring in \mathcal{S}^e are replaced in \mathcal{S}' with respective attribute values characterizing x at least for all objects $I_{\{d\}}(x)$.
- 5) \mathcal{S}' is any completion of \mathcal{S}^e .

If \mathcal{S}' is a completion of type 1 then $\alpha_{AT'}(x,y)=\alpha_{AT^s}(x,y)$ for each $y \in \mathcal{O}V_{\{d\}}(x)$.

Let \mathcal{S}'' be a completion of type 2. For any \mathcal{S}'' there is a completion \mathcal{S}' of type 1 such that attribute values in the both completions are the same for objects from $\mathcal{O}V_{\{d\}}(x)$. One may notice that $\alpha_{AT''}(x,y)=\alpha_{AT'}(x,y)=\alpha_{AT^s}(x,y)$ for each $y \in \mathcal{O}V_{\{d\}}(x)$.

Let be a completion of type 3. There is only one completion of that type, namely \mathcal{S}^x . We may easily notice that $\alpha_{AT^x}(x,y)=\alpha_{AT^e}(x,y)$ for each $y \in \mathcal{O}V_{\{d\}}(x)$.

Let \mathcal{S}'' be a completion of type 4. Let $Z \subseteq \mathcal{O}V_{\{d\}}(x)$ be a set of objects in which original null values were replaced in \mathcal{S}'' with respective attribute values characterizing object x . For any \mathcal{S}'' there is a completion \mathcal{S}' of type 1 that has the same values as \mathcal{S}'' for all objects $\mathcal{O}V_{\{d\}} \setminus Z$. In such a case $(\overline{ATI}_{\{d\}}(x))' \subseteq (\overline{ATI}_{\{d\}}(x))'' \subseteq (\overline{ATI}_{\{d\}}(x))^x$ and thus $Y_p^x \subseteq Y_p'' \subseteq Y_p'$. We may also notice that $\alpha_{AT''}(x,y)=\alpha_{AT^x}(x,y)$ if $y \in Z$ and $\alpha_{AT''}(x,y)=\alpha_{AT^e}(x,y)=\alpha_{AT^s}(x,y)$ if $y \in \mathcal{O}V_{\{d\}}(x) \setminus Z$.

Let \mathcal{S}'' be a completion of type 5. For any \mathcal{S}'' there is a completion \mathcal{S}' of type 4 that has the same values as \mathcal{S}'' for all objects $\mathcal{O}V_{\{d\}}(x)$. We may easily notice that $\alpha_{AT''}(x,y)=\alpha_{AT^e}(x,y)$ for each $y \in \mathcal{O}V_{\{d\}}(x)$. The result obtained for \mathcal{S}'' is valid for any completion of \mathcal{S}^e . Therefore, we can write:

$$\Delta_p^e(x) = \prod_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} \prod_{y \in \mathcal{V}_p} \sum \alpha_{AT'}(x,y) = \prod_{y \in \mathcal{V}_p} \sum \alpha_{AT^s}(x,y) \wedge \prod_{y \in \mathcal{V}_p} \sum \alpha_{AT^x}(x,y).$$

Let us note that $\bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} (\overline{ATI}_d(x))' \subseteq (\overline{ATI}_{\{d\}}(x))^x$, so $Y_p^x \subseteq Y_p'$. Furthermore, $\alpha_{AT^s}(x,y) \wedge \alpha_{AT^x}(x,y) = \alpha_{AT^x}(x,y) = \alpha_{AT^e}(x,y)$ and $\alpha_{AT^s}(x,y) = \beta_{AT^e}(x,y)$ for each $y \in \mathcal{O}V_{\{d\}}(x)$. Hence,

$$\Delta_p^e(x) = \prod_{y \in \mathcal{V}_p} \sum \alpha_{AT^s}(x,y) \wedge \prod_{y \in \mathcal{V}_p} \sum \alpha_{AT^x}(x,y) = \prod_{y \in \mathcal{V}_p} \sum \beta_{AT^e}(x,y) \wedge \prod_{y \in \mathcal{V}_p} \sum \alpha_{AT^e}(x,y).$$

□

6. Illustrative Example

Table 1 presents an incomplete decision table \mathcal{S} containing information about cars. Attribute domains of \mathcal{S} are as follows: $V_{Price} = \{high, low\}$, $V_{Mileage} = \{high, low\}$, $V_{Size} = \{full, compact\}$, $V_{Max-Speed} = \{high, low\}$, $V_d = \{poor, good, excel.\}$.

Car	Price	Mileage	Size	Max-Speed	d
1	high	low	full	low	good
2	low	*	full	low	good
3	*	*	comp	high	poor
4	high	*	full	high	good
5	*	*	full	high	excel
6	low	low	full	*	good

Table 1. Car table

We will illustrate the method of rules' generation using object 6 as a rule generator. Further on, we will refer to the following sets of objects computed in \mathcal{S} : $I_{\{d\}}(6) =$

$$\|(d,good)\|=\{1,2,4,6\}, S_{AT}(6)=\{2,5,6\}, \overline{ATI}_{\{d\}}(6)=\{1,2,4,5,6\}, \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI}_d(6))' =$$

$\{1,2,4,6\}$ (see Property 3.2.2). There are two possible complete descriptors of object 6:

1. $(P,low)\wedge(M,low)\wedge(S,full)\wedge(X,low)$,
2. $(P,low)\wedge(M,low)\wedge(S,full)\wedge(X,high)$,

where P, M, S, X stand for *Price, Mileage, Size* and *Max-Speed*, respectively.

Since $S_{AT}(6)\subseteq \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}} (\overline{ATI}_d(6))'$, then possible rules supported by object 6 may be

generated regardless its possible complete descriptor (see Proposition 5.3.3). The feasibility of generating certain rule covered by object 6 will depend on a considered possible complete descriptor of the object.

Let \mathcal{S}^e be an extension of \mathcal{S} in which object 6 has full complete descriptor as in the case 1 and \mathcal{S}^e differs from \mathcal{S} only for object 6.

- Since $S_{AT}^e(6)=\{2,6\}\subseteq I_{\{d\}}(6)$, then certain rules may be generated from the descriptor of 6 (see Proposition 5.1.3). We will apply Proposition 5.2.1 to compute certain reducts: $Y_C = \emptyset \vee I_{\{d\}}(6) = \{3,5\}$, $\alpha_{AT}(6,3) = \{S, X\}$, $\alpha_A(6,5) = \{X\}$, so $\Delta_C^e(6) = (S \vee X) \wedge (X) = X$. Thus, there is only one certain reduct $\{X\}$ found for object 6 in extension \mathcal{S}^e , which means that only one certain rule is supported by object 6 in \mathcal{S}^e , namely: $(X,low) \rightarrow (d,good)$.

- We will apply Proposition 5.4.1 to compute possible reducts: $(\overline{ATI}_{\{d\}}(6))^x = \{1,2,4,6\}$, $\bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} (\overline{ATI}_d(6))' = \{1,2,4,6\}$ (see Property 3.2.2), so

$$Y_P^x = \emptyset \setminus (\overline{ATI}_{\{d\}}(6))^x = \{3,5\}, \quad Y = \emptyset \setminus \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} (\overline{ATI}_d(6))' = \{3,5\} \quad \text{and} \quad Y \setminus Y_P^x = \emptyset.$$

$\alpha_{AT}(6,3) = \{S, X\}$, $\alpha_A(6,5) = \{X\}$, so $\Delta_P^e(x) = (S \vee X) \wedge (X) = X$. Hence we receive one possible rule: $(X,low) \rightarrow (d,good)$.

Let \mathcal{S}^e be an extension of \mathcal{S} in which object 6 has full complete descriptor as in the case 2 and \mathcal{S}^e differs from \mathcal{S} only for object 6.

- $S_{AT}^e(6) = \{5,6\}$ is not a subset of $I_{\{d\}}(6)$, so no certain rules may be generated from the descriptor of object 6 (see Proposition 5.1.3).

- We will apply Proposition 5.4.1 to compute possible reducts: $(\overline{ATI}_{\{d\}}(6))^x = \{1,2,4,5,6\}$, $\bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} (\overline{ATI}_d(6))' = \{1,2,4,6\}$ (see Property 3.2.2), so

$$Y_P^x = \emptyset \setminus (\overline{ATI}_{\{d\}}(6))^x = \{3\}, \quad Y = \emptyset \setminus \bigcap_{\mathcal{S}' \text{ is a completion of } \mathcal{S}^e} (\overline{ATI}_d(6))' = \{3,5\} \quad \text{and} \quad Y \setminus Y_P^x = \{5\}.$$

$\beta_{AT}(6,5) = \{P, M\}$, $\alpha_{AT}(6,3) = \{S\}$, so $\Delta_P^e(x) = (P \vee M) \wedge (S) = P \vee M \wedge S$. Hence we receive two possible rules: $(P,low) \wedge (S,full) \rightarrow (d,good)$ and $(M,low) \wedge (S,full) \rightarrow (d,good)$. □

Conclusion

In the paper we defined the notions of a certain and a possible rule in an incomplete information system. According to our definitions a rule is certain/possible in the incomplete information system if it is certain/possible in every completion of the initial systems. We showed how to determine such rules directly from the incomplete

IS by applying e.g. Boolean reasoning. Rules may be generated as prime implicants of some Boolean functions. Hence, the problem is NP-hard. Nevertheless, efficient heuristics like Johnson's approximation strategy or genetic algorithms may be applied if we do not wish to generate all rules, but suboptimal ones with minimal condition parts. It is proved in [9] that suboptimal rules supported by an object may be generated in $O(kn^2)$ or $O(k^2n \log n)$ time, where n is the number of objects and k is the number of attributes. Space complexity of our method of rules' generation is linear with regard to the number of objects in the initial incomplete system. Rules supported by different objects' descriptors may be generated in parallel. The method allows to generate all certain rules. Our method of possible rules' generation is valid also in the case of any object-generator such that all objects similar to it are contained in the upper approximation of the same decision class for each completion of an incomplete IS .

References

- [1] Kononenko I., Bratko I., Roskar E., Experiments in Automatic Learning of Medical Diagnostic Rules, Technical Report, Jozef Stefan Institute, Ljubljana, Yugoslavia, 1984.
- [2] Quinlan J.R., Induction of Decision Trees, in *Readings in Machine Learning*, Shavlik J.W., Dietterich T.G. (ed.), 1990, Morgan Kaufmann Publishers, pp. 57-69.
- [3] Chmielewski M.R., Grzymala-Busse J.W., Peterson N.W., Than S., The Rule Induction System LERS - A Version for Personal Computers, *Foundations of Computing and Decision Sciences*, Vol. 18 No. 3-4, 1993, pp. 181-212.
- [4] Kryszkiewicz M., Rules in Incomplete Information Systems, submitted to *Information Sciences*.
- [5] Kryszkiewicz M., Rules in Incomplete Information Systems, submitted to *Information Sciences* Proceedings from the Third Joint Conference on Information Sciences, North Carolina, USA, March 2-5, 1997, to appear. Kryszkiewicz M., Rough Set Approach to Incomplete Information Systems, Proceedings of *Second Annual Joint Conference on Information Sciences*, Wrightsville Beach, North Carolina, USA, 28 September - 1 October 1995, pp. 194-197.
- [6] Slowinski R., Stefanowski J., Rough-Set Reasoning about Uncertain Data, in *Fundamenta Informaticae*, Vol. 27, No. 2-3, 1996, pp. 229-244.
- [7] Skowron A., Rauszer C., The Discernibility Matrices and Functions in Information Systems, in *Intelligent Decision Support: Handbook of Applications and Advances of Rough Sets Theory*, Slowinski R. (ed.), 1992, Kluwer Academic Publisher, pp.331-362.
- [8] Lipski W.J., On Semantic Issues Connected with Incomplete Information Databases, *ACM Transaction on Databases Systems*, 4, 1979, pp. 262-296.
- [9] Nguyen S.H., Nguyen H.S., Some Efficient Algorithms for Rough Set Methods, in *Proceedings of Sixth Intl. Conference IPMU '96*, July 1-5, Granada, Espana, Vol. 3, 1996, pp. 1451-1456.